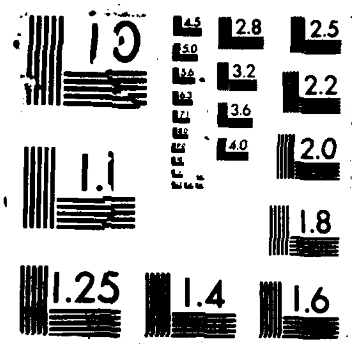
AD-A182 289 VISION ALGORITHMS AND PSYCHOPHYSICS(U) MASSACHUSETTS INST OF TECHNOLOGY CAMBRIDGE DEPT OF BRAIN AND COGNITIVE SCIENCES W A RICHARDS APR 86 1/1

UNCLASSIFIED AFOSR-TR-87-0625 F49620-83-C-0135 F/G 6/4 NL


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

2

DTIC FILE COPY

VISION ALGORITHMS AND PSYCHOPHYSICS
(Final Report 1983–85)

AFOSR-TR- 87-0625

Investigator
(617)253 – 5776 or 5700
Arpanet: WHIT@MIT-MC
Contract No. F49620 – 83 – C – 0135

Whitman A. Richards E10 – 120
Massachusetts Institute of Technology
Cambridge, MA 02139

AD-A182 289

Approved for public release; distribution unlimited.

ABSTRACT: Vision by man or machine is the construction of useful symbolic descriptions from images of the world. Studies of the human visual system provide valuable insights into the kinds of descriptions that will be the most useful, but little insight into the computational problems involved in deriving and manipulating these descriptions. This research examines several computational problems associated with aspects of two- and three-dimensional vision. The solution to these problems includes the design and implementation of particular algorithms. Their efficiency and flexibility is compared with that of the human visual processor.

Key words: Image understanding, visual pattern recognition, vision algorithms, human vision, biological information processing.

April 1986

20 APR 1987

87 5 20 143

Contents

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS N/A
2a. SECURITY CLASSIFICATION AUTHORITY N/A	3. DISTRIBUTION/AVAILABILITY OF REPORT Unlimited--appears in public domain
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) N/A	5. MONITORING ORGANIZATION REPORT NUMBER(S) AFOSR-TR- 87-0625
6a. NAME OF PERFORMING ORGANIZATION Massachusetts Institute of Technology 6b. OFFICE SYMBOL (If applicable) N/A	7a. NAME OF MONITORING ORGANIZATION Air Force Office Scientific Rsch/NL
6c. ADDRESS (City, State and ZIP Code) Dept. of Brain & Cognitive Sciences E10-120, 79 Amherst Street Cambridge, MA 02139	7b. ADDRESS (City, State and ZIP Code) Building 410 Bolling AFB DC 20332-6448
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Air Force Office of Sponsored Research 8b. OFFICE SYMBOL (If applicable) AFOSR/NL	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F49620-83-C-0135
8c. ADDRESS (City, State and ZIP Code) Bolling Air Force Base Washington, DC 20332	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO.: 61102F PROJECT NO.: 2313 TASK NO.: A5 WORK UNIT NO.:
11. TITLE (Include Security Classification) VISUAL ALGORITHMS AND PSYCHOPHYSICS	

12. PERSONAL AUTHOR(S)
Richards, Whitman A.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
Final	FROM 1983 TO 1985	1986 April	24

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	Image understanding, visual pattern recognition, vision algorithms, human vision, biological information processing

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Vision by man or machine is the construction of useful symbolic descriptions from images of the world. Studies of the human visual system provide valuable insights into the kinds of descriptions that will be the most useful, but little insight into the computational problems involved in deriving and manipulating these descriptions. This research examines several computational problems associated with aspects of two- and three-dimensional vision. The solution to these problems includes the design and implementation of particular algorithms. Their efficiency and flexibility is compared with that of the human visual processor.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Dr. John Tangney	22b. TELEPHONE NUMBER (Include Area Code) 202-767-5021 22c. OFFICE SYMBOL AFOSR/NL

Vision Algorithms and Psychophysics

1.0 Introduction: The Problem and Goal

"Seeing" requires the construction of symbolic descriptions of the external world. The most useful symbolic descriptions will be representations for each of the various objects in the three-dimensional scene. These objects, in turn, may be broken down further into more detailed modular representations that may include the various attributes of each object such as its color, texture, or the shape and relative motion of its parts. These latter properties are thus our basic building blocks from which more complicated descriptions are built. Vision understanding requires showing how such object properties can be represented internally, and how they can be brought together to create a description suitable for recognition or manipulation. This then, is our ultimate goal: to propose and implement a scheme for representing 3D shapes in a manner suitable for recognition.

To move toward this goal, the research has proceeded along several parallel tracks. The first is the development of a theory for representing 3D shapes, or their 2D projections onto the image. Such a theory requires first that the shape be spatially isolated. Hence the second research track is the identification of object candidates, using texture (scale-space) and visual motion (or stereo). A third track is a machine implementation of the proposed schemes. And finally, a fourth research area interwoven among the others is the psychophysical explorations that provide hints about viable vision algorithms.

2.0 The Codon Theory and Its Implementation

Shape is one of the most important ways of categorizing and identifying objects. In Figure 1, the very simple shape of an eye immediately implies "animal". Seeing the "beak" would further constrain the class to be "bird". In contrast, an isolated patch of texture is almost meaningless. This simple observation shows that rather simple shapes can provide a powerful representation for recognizing objects. Silhouettes or cartoons reinforce this notion. What, then, are the basic elements from which we can build simple shapes and make such powerful inferences from image contours?

In 1982, Hoffman & Richards proposed a primitive representation for the shape of 2D or plane curves. The key concept was that the representation should

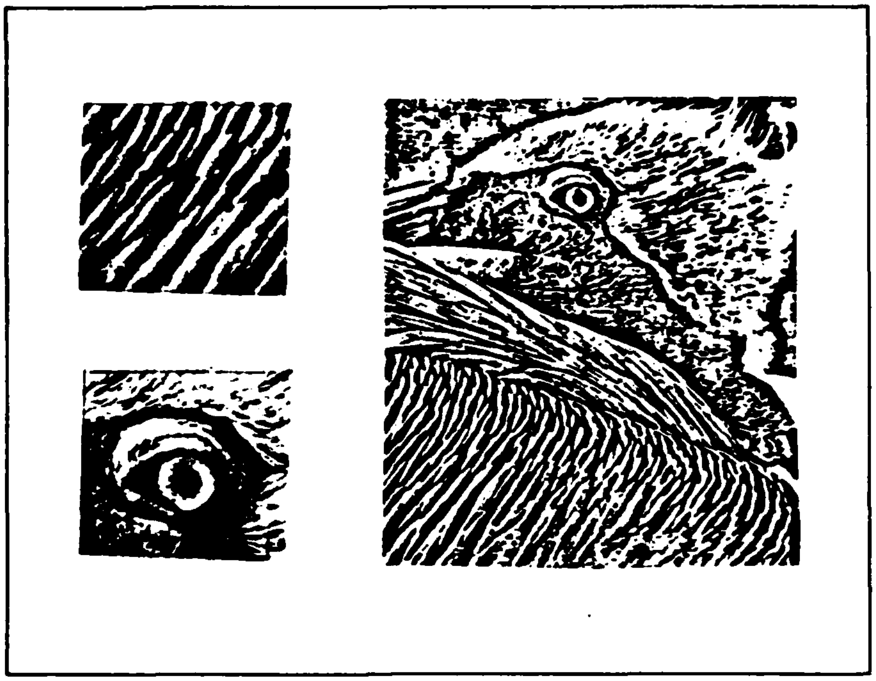


Figure 1 The left-hand panels show two portions of the bird—one a texture, the other a "shape". Clearly the simple shape of an eye alone provides an important pointer to the class of object, namely "animal", whereas the texture patch alone offers few clues.

make explicit the parts of a shape (or 3D object), because objects are described most naturally in terms of their "parts".

To find the parts of an object or even of a plane curve, one notes that when two parts are joined, a concavity in the surface will be formed, which appears as a minima of curvature or cusp in the image (see Figure 2). This concavity property is a transversal one—stable under perturbations of the way the parts may be joined. Transversality is thus a fundamental regularity of natural objects. The resulting concavities in the 2D silhouette are known to be visually important (Atteneave, 1954; Biederman, 1984). Consequently both psychophysical and computational arguments underlie our scheme for segmenting a shpe into "parts" for recognition.

To represent the shape of the "parts" isolated by the concavities, or minima of curvature, Hoffman & Richards (1982, 1983, 1984) propose as a first abstract description that we use the singularities of curvature. These are the maxima, minima and zeroes of curvature along a plane curve. Such a representation has the

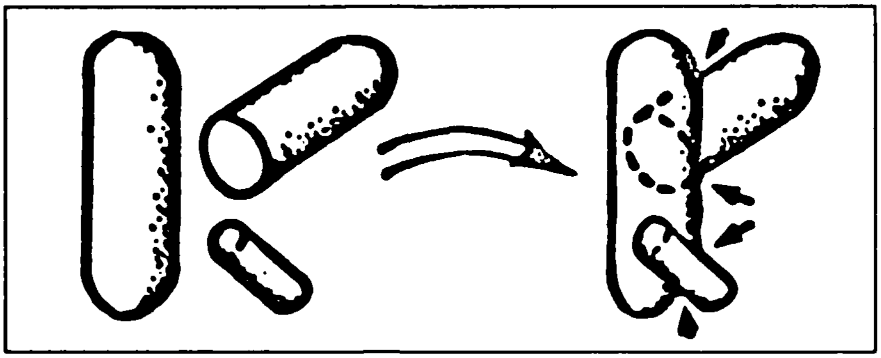


Figure 2 Joining parts generally produces concavities in the silhouette.

important feature that it is invariant under similarity transforms—rotation, translation or dilation. The basic elements of the representation are called "codons", which are illustrated in Figure 3. Each codon simply represents one of five possible relations between the maxima, minima and zeros of curvature. They are identified by their number of inflections (zeroes). Shapes described in terms of a sequence of codons have several interesting formal properties, such as making skewed symmetry explicit, and being sensitive to the choice of figure and ground (see Figure 4)—two important perceptual attributes (Hoffman & Richards, 1982, 1984).

Over the past few years, the research on the codon scheme for representing shapes has focused on four problems: (a) a rigorous mathematical statement of the transversality motion and its generalization to smooth shapes, (b) topological constraints on possible smooth 2D (image) contours defined by codon strings, (c)implementation, and (d) predicting 3D shape from a 2D (image) contour.

First we will summarize some work on the transversality regularity that is critical to a formal definition of a "part boundary".

2.1 Transversality and "Parts"

When two surfaces intersect they intersect *transversally*. This means that the tangent planes to the two intersecting surfaces are of different orientations at each point where they intersect, implying that there is a discontinuity of the tangent plane to the surface of the new composite object at each point along the contour

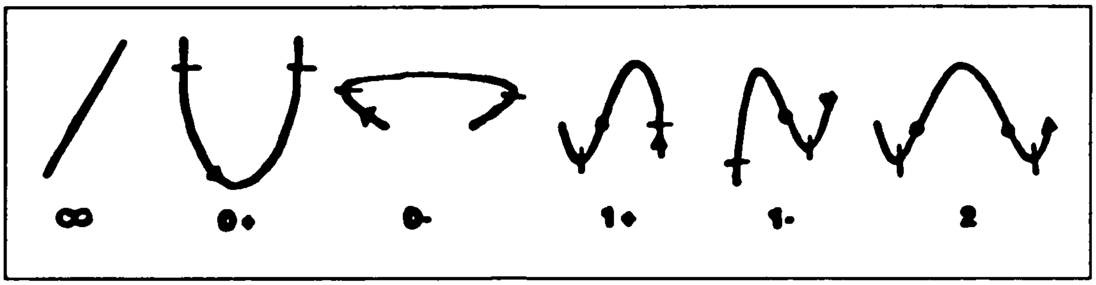


Figure 3 The primitive codon types. Zeroes of curvature are indicated by dots, minima by slashes. The straight line (∞) is a degenerate case included for completeness, although it is not treated in the text. (See Richards & Hoffman, 1984, for definitions.)

of intersection (see Figure 2). Contours of concave discontinuity are thus the part boundaries.

Consider now what happens if the concave discontinuity at the part boundary is smoothed as if a membrane were stretched across the discontinuity in the intersecting surfaces. Where then is the part boundary on the smoothed surface? Intuitively, one might choose the locus having greatest curvature. Hence we have proposed the following rule for partitioning smooth surfaces into parts:

Negative Minima Partitioning Rule: Divide a surface into parts at negative minima of the principal curvatures along their associated lines of curvature.

The proof that this rule indeed captures our intuitive notion of the part boundary is quite difficult, but has recently been completed by Bennett & Hoffman (1986). The thrust of the proof is to show that the smoothing of a concave discontinuity on a surface will produce local extremum of surface curvature in the neighborhood. The proof thus provides a solid mathematical foundation to our part boundary notion for 3D shapes. Given this mathematical rigor, we can now determine how the 3D boundary will appear when projected into the image. Although the boundary will generally *not* be a point of extremal curvature, we expect that 2D extrema of (negative) curvature will lie near the projection of a 3D part boundary. The most common case where the projection has a cusp in the occluding contour has already been treated by Hoffman & Richards (1984).

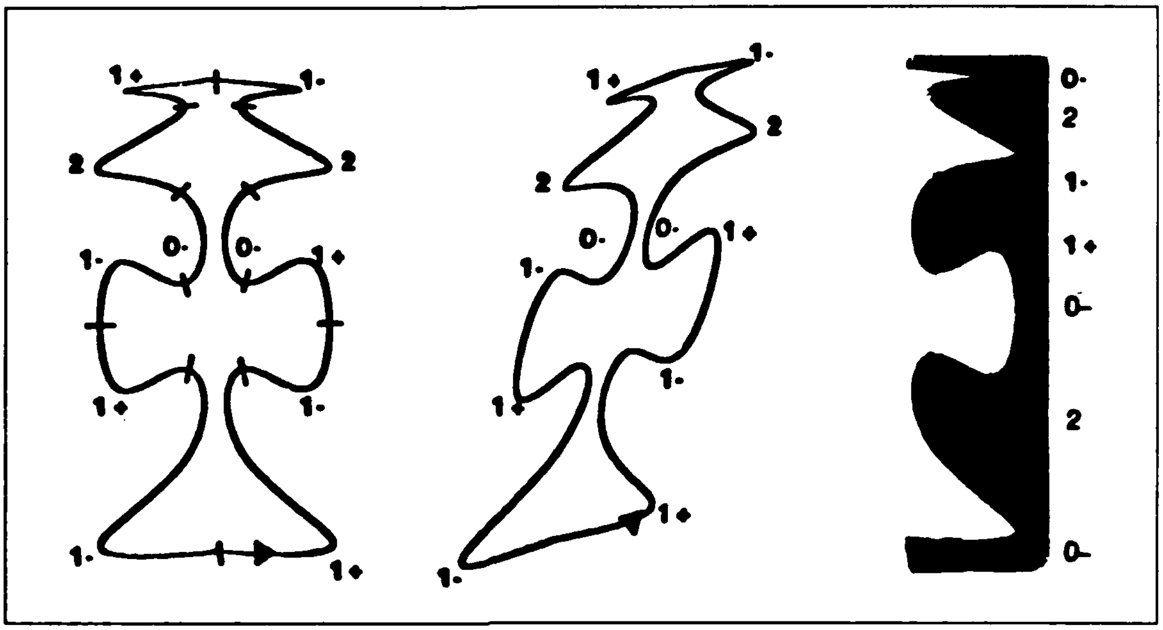


Figure 4 Skewed symmetry is obvious in the codon string because half the sequence is reversed, ignoring the sign of the codon (left frame). Figure-ground reversal changes the codon string because maxima and minima of curvature are exchanged, providing a simple explanation for Rubin's face-vase illusion.

2.2 Constraints on Codons

Clearly any plane curve can be described by a sequence of codons, for all curves can be characterized by a sequence of the extrema of curvature. However, once one imposes restrictions on the behavior of a plane curve, the sequence of curvature extrema may not be arbitrary. One example is the class of plane curves that are smooth and have no cusps. Included in this class are the smooth plane curves that represent the canonical outlines of smooth 3D shapes.

To see that not all sequences of codons are possible if the curve is smooth, refer to Figure 3 once again. Note that a 1^- can not follow a 1^- codon unless a cusp is allowed. Similarly, a 1^+ can not follow a 1^+, because if such a join is attempted either a cusp will be created or, if the curve is indeed smooth, the 1^+ codon would have to be transformed into a type 2. To specify all legal smooth codon strings, we will first enumerate all pairs, and then show what pair substitutions are legal for one element in a sequence of pairs, thereby creating all possible triples.

LEGAL CODON PAIRS			THIRD CODON									
			0^-		0^+		1^-		1^{+-}		2	
			−	−	+	+	−	+	+	−	−	−
	TAIL	HEAD	TAIL	HEAD	TAIL	HEAD	TAIL	HEAD	TAIL	HEAD	TAIL	HEAD
0^-0^-	−	−	[+	+]			+			+	[+	+]
0^-1^-	−	+		+	+				[+	+]		+
0^-2	−	−	[+	+]			+			+	[+	+]
0^+0^+	+	+			[+	+]		+	+			
0^+1^+	+	−	+			+	[+	+]			+	
1^-0^+	−	+		+	+				[+	+]		+
1^-1^+	−	−	[+	+]			+			+	[+	+]
1^+0^-	+	−	+			+	[+	+]			+	
1^+1^-	+	+			[+	+]		+	+			
1^+2	+	−	+			+	[+	+]			+	
$2\ 0^-$	−	−	[+	+]			+			+	[+	+]
$2\ 1^-$	−	+		+	+				[+	+]		+
$2\ 2$	−	−	[+	+]			+			+	[+	+]
NUMBER OF LEGAL PAIR SUBSTITUTIONS			5		2		3		3		5	
NUMBER OF PAIR SUBSTITUTIONS			9		4		6		6		9	

Note. The third codon can either follow or precede the pair. A (+) indicates a proper join. Because of symmetry, there are an equal number of total pluses in the head and tail columns.

Figure 5 Table 1: Legal smooth codon triplets. The third codon can either follow or precede the pair. A (+) indicates a proper join. Because of symmetry, there are an equal number of total pluses in the head and tail columns.

Define the "tail" of a codon as the region about the first minima encountered when traversing the curve. The "head" of the codon is the subsequent minima. A smooth string of two codons is then allowable only if the head of the first codon has the same sign of curvature as the tail of the second codon in the string. To enumerate the possible codon pairs for a smooth contour, we require that the curvature of both the head and tail of a middle codon match the tail of its successor or the head of its predecessor in the string. All such legal pairs are given

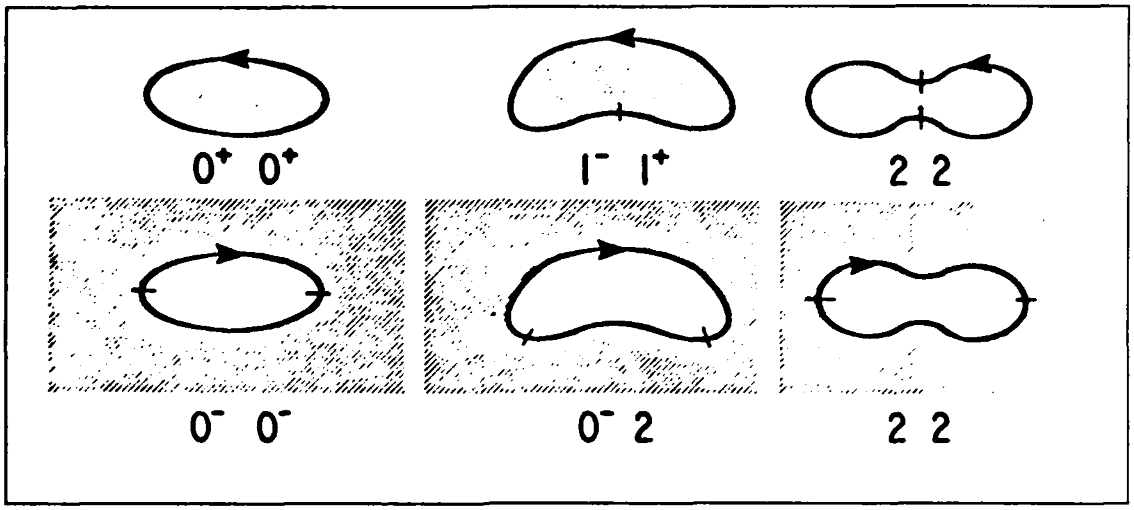


Figure 6 Legal smooth, closed codon pairs. Figure is indicated by cross hatching. Part boundaries are noted by the slashes.

in the left column of Figure 5. There are only 13 legal pairs out of a possible 25 combinations.

If we now require that the curve be closed smoothly on itself, then this constraint drastically reduces the number of legal pairs, for now the head and tail of the pairs must have the same sign. Inspecting Figure 5, we see immediately that only $0^- 0^-$, $0^- 2$, $0^+ 0^+$, $1^- 1^+$ and 2 2 qualify. These shapes are shown in Figure 6. Surprisingly, now there are only three such legal shapes out of the possible 25 combinations! According to codon theory, the "ellipse", the "peanut" and the "dumbbell" are the three most primitive shapes. Note that the ellipse has no "parts", the peanut has one part boundary, and the dumbbell has two parts. It will be these three primitive shapes that our implementation will seek in images, to be described in a later section.

In a similar manner, we can enumerate the class of all possible smooth-closed shapes that are topologically similar in their "bumps" and dents". There are not very many forms of such curves. For example, a seven codon sequence has 78,125 possible combinations, but only 96 are allowable (see Richards & Hoffman, 1984, for proof). Figure 7 shows legal smooth, closed plane curves for codon strings of length three and four. Note that even four codon elements can yield quite descriptive-looking shapes, such as the "fetus" or "animal" in the lower right. Furthermore, it should be obvious that combinations of closed codon shapes

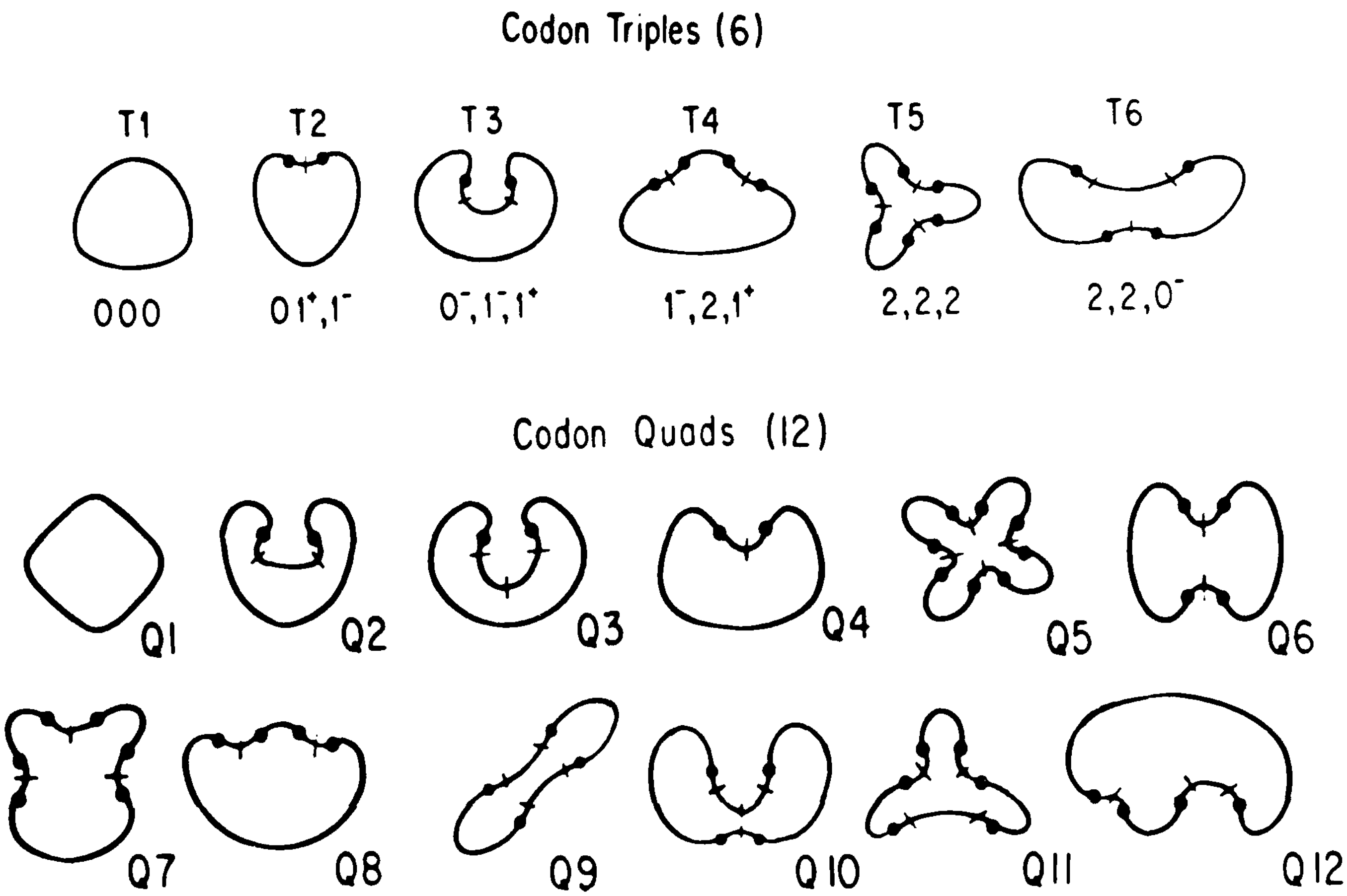


Figure 7 Legal smooth, closed codon triple and quadruples.

embedded within one another can represent a wide range of complex figures. The "eye" of Figure 1, which is simply one ellipse within another, is one example. Or a "face" which is often depicted by an ovoid with two simple ellipses for the "eyes" and a "peanut" shape for the mouth, would be another example. How, then, can the codon shapes be extracted from images in order to build such representation?

3.0 Codon Implementation

3.1 Description Algorithm

In order to represent codons within codons, such as in the eye or face example, we Gaussian filter our images at several scales, using a pyramid scheme proposed by

Figure 8 A Gaussian pyramid structure using the mask shown on the left produces a "pyramid" of images (also shown on the left). These images are used to obtain the two binary pyramids of image "Snoopy", as shown in the right panel.

Burt and Adelson (Burt, 1982; Burt & Adelson, 1983). Figure 8 shows the output of this first stage of processing of the image "Snoopy". Note that the algorithm gives us two pyramids—one capturing the "dark" blobs, the other the "light" blobs. We now are able to create a blob hierarchy, where blobs within blobs are specified as a linked-list tree-structure. Because Gaussian masks are used, we are guaranteed that our blob hierarchy will be well behaved (Koenderink, 1984; Yuille & Poggio, 1984).

With all the blobs located (some in more than one pyramid level), we next generate an edge list for each blob. Starting at the top of the blob, we encode the edge in a counter-clockwise fashion. Our algorithm is an adaptation of a standard edge crawl, using 8-way connectivity. Between a pixel and the previous pixel in the edge list there is a "tangent" – one of eight directions. To find the next pixel in the sequence we project a vector normal to this simple tangent vector (90 degrees clockwise rotation) and then sweep this vector counter-clockwise until it finds the next pixel. In this fashion we "hunt" between object and background and generate the edge list. The implementation is simplified by using only the eight possible vectors in an 8-connected area around the pixel.

At this point a tangent could be computed at each edge point using a standard set of edge masks. Since we are dealing with "blobs" it makes more sense to compute the principal normal. This also turns out to be computationally efficient. Whereas the tangent points along the edge of the blob, the principal normal points towards the center. Consider then an edge point and the points immediately surrounding it. Each surrounding point that is part of the blob (signal) can be thought to pull on rubber bands attached to an imaginary vector emanating from the edge point. The sum of these pulls will tend to point the vector towards the center of the blob, and hence the vector will approximate the principal normal. The computational scheme for implementing this algorithm is described in Dawson & Treese (1984). The output of the algorithm is thus the normal to the outline of the blob. The tangent at each point is simply 90° to this orientation. Figure 9 shows one example for the "dumbbell"-shaped eyebrow of Snoopy.

Because the codon scheme is based upon extrema of curvature, we now must differentiate the tangent versus arc length along the blob outline. In the upper right panel of Figure 9 the tangent versus arc length is given by the upper curve of the graph. Its derivative is obtained simply by applying the "edge" operator shown in the same panel to obtain curvature versus arc length (lower curve on graph). The extremities of curvature used to specify the codons and hence the shape of the blob can now be read off directly (see Dawson & Treese, 1984).

3.2 Sketch of the "See" Machine

A large part of the effort over the past three years was devoted to software and hardware development. Our "See" machine is a VAX 11/750 computer with a 400 megabytes fixed disc, linked to an Adage 3000 image processor. The Adage has 512 x 512 x 24 bits resolution and is our principal image processing device, performing the Gaussian pyramid convolutions in about 17 seconds. It is also used to generate "natural" images. Other peripherals include several single-frame graphics terminals, a matrix color camera, a Fairchild CCD camera and several vidicons eventually intended for color and stereo input. Still under development is an Ethernet connection to the Artificial Intelligence lab, and the capability for inputting motion sequences taken with a portable video camera.

The VAX runs Unix 4.2. Over the past year much special purpose software has been written to make the "See" Machine a user-friendly system suitable for both graphics and implementation of the codon scheme. Over one man-year has

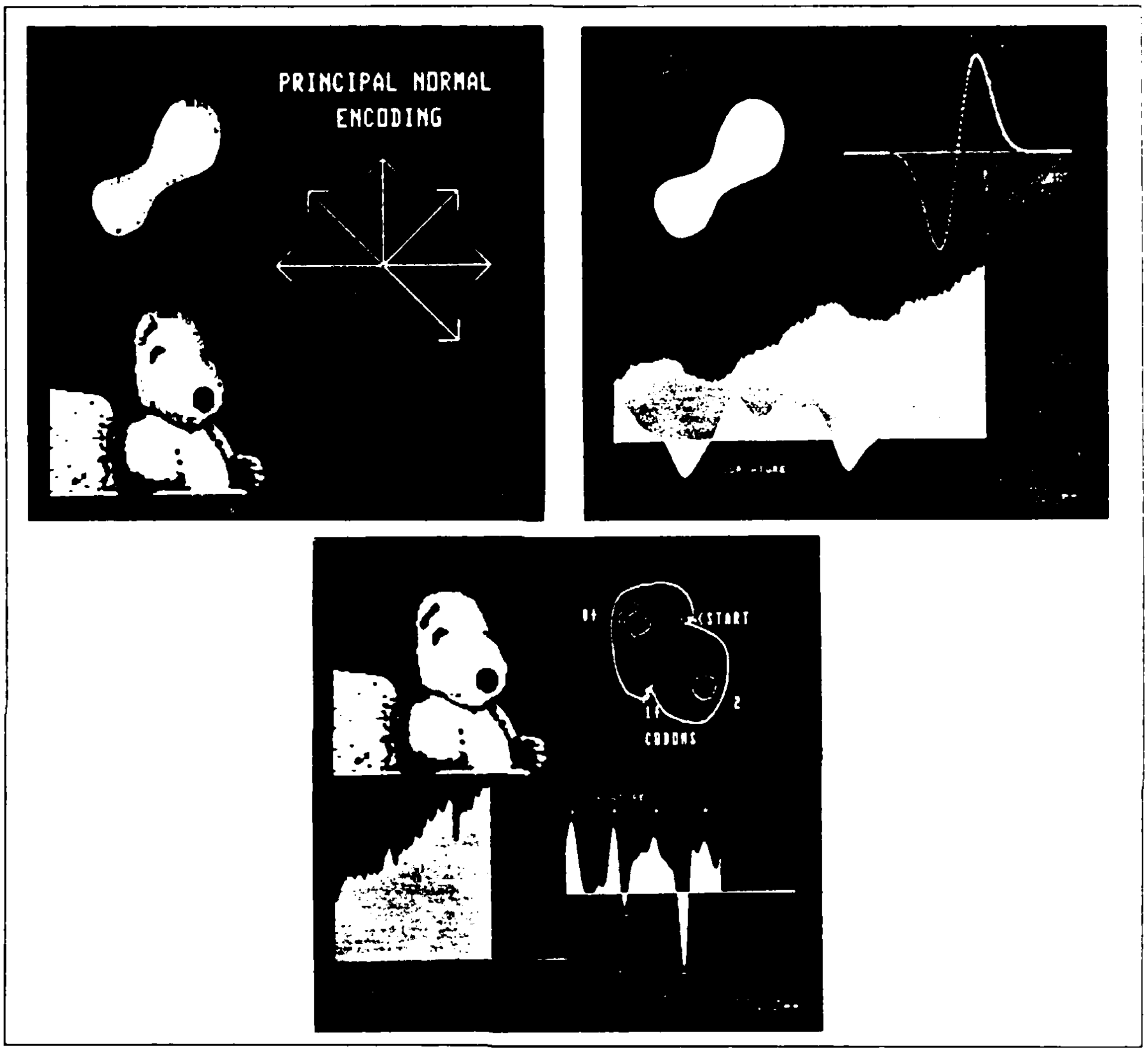


Figure 9 The upper left panel illustrates the principal normal encoding scheme for the sub-blob of Snoopy's eyebrow. The upper right shows how curvature along the blob's outline is computed. A second example for the super ordinate blob of Snoopy's head outline is shown in the third panel.

been spent solely on software development (see Appendix 1 for a list of packages written).

4.0 Inferring 3D Shape from 2D Contour

Although an infinity of 3D objects could generate any given 2D shape, we usually infer only one 3D object from its 2D projection. What are the constraints that restrict this infinity of choices? With Jan Koenderink, we have been studying this problem. Our aim is to predict the rough topology of the perceived 3D shape, given a silhouette such as those illustrated in Figure 7. Specifically, we wish to specify the Gaussian curvature of the inferred 3D surface (Hilbert & Cohn-Vossen, 1952). Koenderink & van Doorn took a major step in this direction in 1976, when they proved that the sign of the Gaussian curvature of points on the 3D surface is the saem as the sign of curvature of their projections into the silhouette. This theorem greatly restricts the class of inferred 3D surfaces, but by itself is not powerful enough to specify a unique 3D shape. A second constraint is an interpretation rule that we have been exploring:

3D Shape Interpretation Rule: Do not propose undulations of the 3D surface without evidence for such.

The above rule is an extension of the "general position" restriction, which requires that the view of an object is not a special one and is stable under perturbation. For our purposes, the restriction states that a slight shift in viewpoint should not change the topology of the viewed structure, such as suddenly revealing a bump or dent in the surface that was previously hidden by occlusion. This interpretation rule, together with the above mathematical property, seems to be the primary forces that drive our interpretation of silhouettes (Richards, Koenderink & Hoffman, 1985).

5.0 Groupings and "Glue"

Most attempts to interpret images focus upon image contours. Hence a large part of image processing is concerned with edge detection algorithms. Our early attempts to isolate shapes also proceeded in this manner, using an algorithm called "CARTOON" (Richards et al., 1982). Such edge-finding schemes when used on natural images are almost guaranteed to yield broken contours, which must be subsequently "glued" together appropriately. One method for identifying the pieces of contour that should be linked is to provide color or texture labels.

Our current scheme which is based upon "blobs" does not suffer this problem, for all "blobs" are guaranteed to have closed boundaries. Nevertheless, sometimes

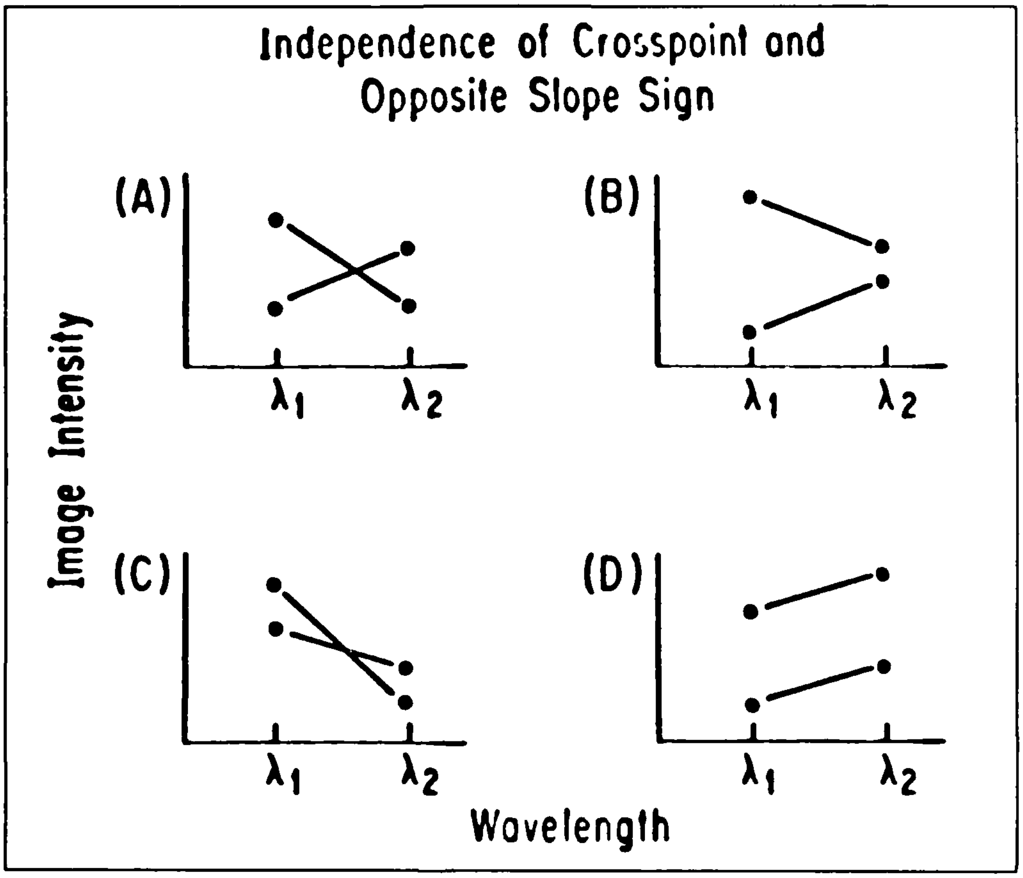


Figure 10 Graphs of image intensity (ordinate) versus wavelength (abscissa). Two wavelength samples, λ^1 and λ^2, are shown. An image region yields two samples of intensity, one for each wavelength, and is represented by the line segment connecting the two sample values. a) & c) Two examples of the spectral crosspoint (Rubin & Richards, 1982). a) & b) Two examples of the opposite slope sign condition. This is the minimal configuration that shows different ordinalities. Note that the crosspoint and opposite slope sign condition are completely independent, since they can occur together (a), or each can occur alone (b and c), or neither can occur (d).

different parts of the same object will appear as isolated blobs (such as the two eyes in a face or during occlusions) and it is useful to be able to assign material-property labels to the isolated blobs to provide indices for appropriate groupings. Color, texture and motion are the prime candidates for such labels.

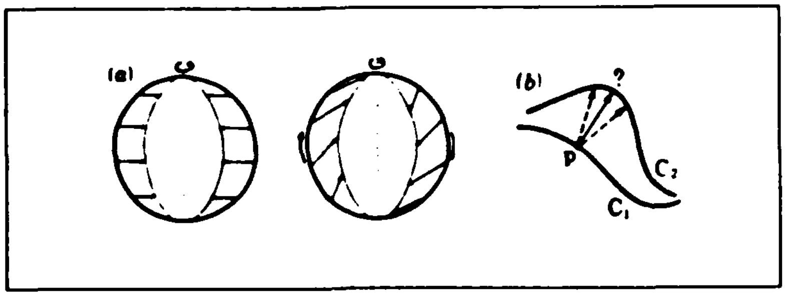


Figure 11 Ambiguity of the velocity field. (a) the arrows represent two possible velocity fields that are consistent with the changing image. (b) The curve C_1 rotates, translates and deforms over time to yield the curve C_2. The velocity of the point p is ambiguous.

5.1 Using Spectral Information to Represent Material Categories.

Earlier, Rubin & Richards (1982) had shown how an operator called the spectral cross-point could be used to find material changes across an edge. This condition is depicted in Figure 10. Also shown in the same figure is a second condition, called the opposite-slope sign, which can be used to categorize the spectral composition of surfaces. In brief, this new condition describes the crude spectral shape of a pigment in terms of its derivatives of absorption versus wavelength. Details are given in a MIT A.I. Lab. Memo 764 by Rubin & Richards (1984). The theory has implications for both psychology and neurophysiology. In particular, Hering's notion of opponent colors and psychologically unique primaries, and Land's results in two-color projection can be interpreted as different aspects of the visual system's goal of categorizing materials. Also, the theory provides two basic interpretations of the function of double-opponent color cells described by neurophysiologists.

5.2 Texture Fields

We have begun some work on representing the local texture properties of a "blob" in terms of the four components of a linear flow field (i.e. dilation, rotation, shear and deformation). This approach is new because it attempts to infer the local organizations of the texture directly without first establishing correspondence (Richards, 1984). There are some formal similarities to Ullman's work in structure-from-motion which are also being explored.

6.0 Visual Motion

Being able to compute the movement in the changing retinal image is critical to an implementation of the codon theory, for it is perhaps the single most powerful method of isolating "blobs" of greatest interest. Such a description of the movement is not provided to our visual system directly, however; it must be inferred from the pattern of changing intensity that reaches the eye. Hildreth (1984a,b) has studied this problem in detail.

One serious obstacle to computing general motion is that the local motion measurements generally provide only one component of the local velocity (this is the aperture problem illustrated in Figure 11). To recover the full velocity field requires constraining the possible range of solutions. Hildreth proposes a smoothness constraint, which is based on the physical assumption that surfaces are generally smooth. A theoretical analysis of the conditions under which these assumptions yields the correct velocity field has been completed (Hildreth, 1984). An algorithm has also been devised and implemented, using several examples that permit comparison with psychological observations. Examples of particular interest are the rotating spiral and the barber pole illusion, where the true velocity vectors are not seen either by the algorithm nor by the human observer (Figure 12). Such failures consistent with human perception give credibility to Hildreth's formulation, suggesting it may be the basis for motion analysis in the most powerful vision machine known—our own!

7.0 Summary

All plane curves may be described by a linked list of codons. For smooth, closed plane curves, the codon sequences enable us to enumerate shapes of increasing complexity. The "ellipse" "peanut" and "dumbbell" are the simplest shapes according to the theory. Even these simple shapes, when embedded in a hierarchy, provide very powerful indices for object recognition.

A nice feature of the codon theory is that the descriptions are computable from images, as we have shown. Current work in progress is the creation of a completely automatic package that will deliver a hierarchy of codon descriptions for a 512×512 static image. At a later date we will add a motion capability for single blob isolation along the lines Hildreth has proposed.

At present, theoretical work has been concentrating on what assertions about 3D shape can be made from the 2D codons. Metric information required for a more detailed description of a "part" is also being considered. Possibilities in-

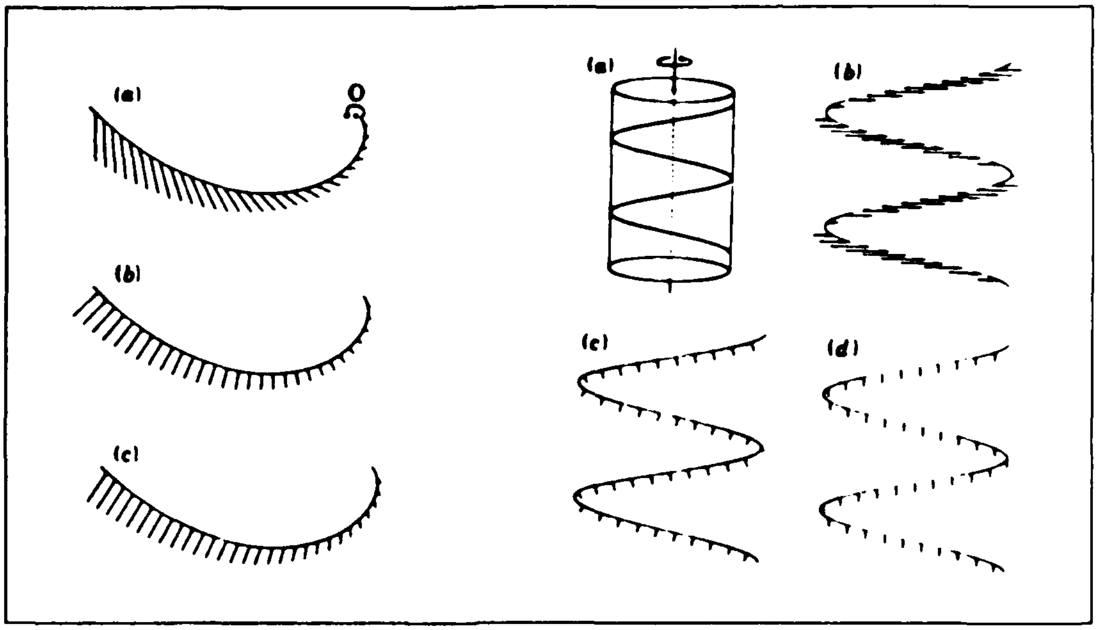


Figure 12 (a) **Left:** Rotating spiral. (a) The true velocity field for a logarithmic spiral rotating in the image about the point O. (b) the initial perpendicular velocity vectors. (c) The computed velocity field of least variation. (b) **Right:** The barberpole illusion. (a) A circular helix on an imaginary cylinder, rotating about the vertical axis of the cylinder. (b) The two-dimensional projection of the helix and its velocity field. (c) The initial perpendicular velocity vectors. (d) The computed velocity field of least variation.

clude abstracting qualitative descriptors of part-shapes such as as "knob", "neck", "bump", "dent", "fold", "finger", etc. The relations between blobs and their sub-blobs must also be made explicit. However, at present we have little insight how this should be done, with the exception of Ullman's paper on "Visual Routines", together with some crude psychophysical observations that we have tentatively begun to explore. Over the next few years, therefore, our primary efforts will be in three areas: 1) supplementing the codon description with axial-based information; 2) using the primitive codon features for stereo and motion matching, and 3) curvature psychophysics—determining the psychological basis for extracting curvature, allowing a comparision between computational and biological approaches encoding 2D shape.

8.0 Appendix 1: Software Developed

Over 200 programs have been written so far for the work of this grant. The programs range from simple subroutines to major packages and systems programs. Here we list the major classes of programs and describe a few examples.

Packages

PYRAMID: This code generates the Gaussian Pyramid from an input image. We have three versions of this code. The original version was developed on a Symbolics 3600 Lisp machine, the next version on our VAX, and the final version is in Adage 3000 BPS microcode. Since this routine is basic to our work work, we feel that the speed improvement (a factor of 10) in the final version was worth the effort.

BLOB: A package for generating the first level symbolic blob descriptors from a Gaussian Pyramid. The images are made into binary images, and these are used to generate a tree structure of blobs. A blob is represented by its location, size, outline, tangents, and pyramid level. Additional constraints, for example color and motion, on the definition of the blobs will be used to reduce the number of blobs and improve their definition.

CODON: This package takes the edge list generated by the BLOB package and generates a Codon description. It differentiates the curve tangents, and applies algorithms to define extrema of curvature. Additional programs are used for display, test, and rudimentary matching of symbolic descriptions.

CARTOON: We originally wrote the CARTOON package on the PDP 11/23. This re-write of the package takes advantage of the VAX's increased performance. The CARTOON package has a user-friendly interface and allows users to manipulate images by doing convolutions, thresholds, color coding, acquiring, saving, and restoring, etc.

FRACTALS: This package allows the generation of a limited class of fractal patterns. We are interested in these patterns as potential descriptors of form and texture in complex images.

OHIO: We have modified the Ohio Rendering Package provided by Prof. Frank Crow to run on the Adage 3000. This package allows us to generate realistic objects with shaded surfaces, multiple light sources, etc. Used for studying the perception of objects.

SPHERES: A package for generating spheres with various shading, reflectance, and light-source functions. Used for studying the perception of reflectance and specularity.

Systems Programs

We have written drivers, linkers and controllers to effectively use the new hardware acquired in part on this grant.

IK: This is the Adage 3000 (Ikonas) system driver. This is a modification of code originally provided by Ron Gordon of Bell Laboratories. Our code is now a distributed Berkeley 4.2 Unix driver for this machine.

RILMOD: We use the GIA/RIL microcode assembler for the Adage 3000 (provided by the University of North Carolina). Extensive modifications of the RIL linker have been made to improve performance and support our hardware.

MC68: Code to utilize the Motorola 68000 processor on the Adage 3000. This enables us to load and run programs, handle interrupts and access the other hardware on the Adage from the 68000.

Slave Machines

Our PDP-11's may be used either as stand-alone machines for image collection and processing, or as "slave" machines performing specific functions under the direction of the VAX.

SEND11, GRAB11, IMAGE11: Allow the VAX to control a PDP-11 used as a remote graphics and image processor.

SLAVE: A remote graphics and image processing that runs on the PDP-11's. Performs such functions as drawing lines, text, circles, etc., acquiring and displaying images.

Libraries

A variety of libraries have been written to support the research program. A library is a collection of routines that serve a common goal.

LIKNEW: The New Ikonas library contains routines for controlling the Adage 3000, graphics (lines, circles, blocks, text, etc.), image acquisition, save and restore, etc. This library contains over 60 subroutines, and includes an extensive shared data base for specification of the Adage parameters.

GRAPHICS: A collection of simple and advanced graphics routines written in portable C code so they can be used on a variety of machines.

USEFUL: A collection of useful tools and routines. Used in most of our programs.

Communications

Our VAX is linked (currently via RS-232 lines) to the main AI machine (OZ) and our "slave" PDP-11's. These connections will be replaced by Ethernet links.

UURT, RTU and UNET: Communication and file transfer between the VAX and a PDP-11. Allows a PDP-11 to appear as a "virtual" VAX terminal. This is an extensive re-write of code provided by the Center for Cognitive Science at M.I.T.

PR: Remote printing of text files on various printers.

OZLINK, OZTALK: Communication and file transfer (using the KERMIT protocol) between the VAX and the OZ (AI) machine.

Tests and Demonstrations

TEST programs include programs for testing the Adage 3000's various components, for testing the communication links, and for testing developing code. The DEMONSTRATION packages link components and other packages to demonstrate such things as the entire blob/codon system and the graphics capability of the Adage 3000.

Appendix 2: Publications and References

Attneave, F. (1954) Some informational aspects of visual perception. *Psychol. Rev.*, 61:183–193.

Bennet, J.B. & Hoffman, D. (1985) Shape decomposition via transversality. In *Image Understanding 1985-86*, (eds.) W. Richards & S. Ullman. Norwood, NJ:Ablex.

Biederman, I. (1984) Recognition by components: a theory for image interpretation. Workshop on Human and Machine Vision, Montreal, August 1–3. To appear in *Comp. Graphics & Image Processing.*

*Bobick, A. (1984) Grouping visual targets. Abstract. Perceptual Organizations Workshop, Pajaro Dunes, CA.

*Bobick, A. (1984) Using mirror reflections to recover shape. Abstract. WH2, OSA Annual Meeting.

Burt, J.B. & Adelson, E.H. (1983) The Laplacian Pyramid as a compact image code. *IEEE Trans. on Comm.*, Vol. COM–31, No. 4, pp. 532–540.

Burt, J.B. (1982) The pyramid as a structure for efficient computation. Rensselear Polytechnic Institute Image Processing Laboratory Technical Report IPL–TR–038.

*Dawson, B.M. & Treese, G. (1985) On finding objects in images. *SPIE Conference on Image Processing.*

*Dawson, B.M. & Treese, G. (1984) Computing curvature from images. *Proceedings SPIE*, vol. 504.

*Gilson, W. (1984) Idiot's guide to OZ (A manual for text editing). *Natural Computation Group.*

*Hildreth, E.C. (1984) Computations underlying the measurement of visual motion. MIT A.I. Lab. Memo 761.

*Hildreth, E.C. (1984) Computing the velocity field. Abstract. TuC2, OSA Annual Meeting.

*Hildreth, E.C. (1984) *The Measurement of Visual Motion.* MIT Press, Cambridge, MA.

*Supported by current AFOSR contract.

*Hildreth, E.C. (1983) The computation of the velocity field. Massachusetts Institute Of Technology A.I. Lab. Memo 734.

Hilbert, D. & Cohn-Vossen, S. (1952) *Geometry and the Imagination.* Chelsea, NY.

*Hoffman, D.D. & Richards, W. (1984) Parts of recognition. Abstract. TuC3, OSA Annual Meeting.

*Hoffman, D.D. & Richards, W.A. (1984) Parts of recognition. *Cognition,* 18:65 - 96.

*Hoffman, D.D. & Richards, W. (1983) Parts of recognition. MIT A.I. Lab. Memo 732.

Hoffman, D. & Richards, W. (1982) Representing smooth plane curves for visual recognition: implications for figure-ground reversal. *Proc. American Assoc. Artificial Intelligence (AAAI-82).*

Koenderink, J.J. (1984) The structure of images. *Biol. Cybern.*, 50:363 - 370.

Koenderink, J.J. & van Doorn, A.J. (1976) The singularities of the visual mapping. *Biol. Cyber.*, 24:51 - 59.

*Richards, W. (1985) Structure from stereo and motion. *J. Opt. Soc. Amer.*, 2:343 - 349.. [Also MIT A.I. Lab. Memo 731 (1983).]

*Richards, W. (1984) Grouping without prior correspondence. Abstract. WP1, OSA Annual Meeting.

*Richards, W. (1984) Interpreting image contours. Abstract. Air Force Vision Program Review, May.

*Richards, W. & Hoffman, D. (1985) Codon Constraints on Closed 2D Shapes. *Computer Vision, Graphics & Image Processing*, 31:265 - 281. [Also MIT A.I. Lab. Memo 769 (1984).]

*Richards, W., Koenderink, J.J. & Hoffman, D.D. (1985) Inferring 3D shapes from 2D codons. MIT A.I. Lab. Memo 840.

Richards, W., Nishihara, H.K. & Dawson, B. (1982) Cartoon: a biologically motivated edge detection algorithm. MIT A.I. Lab. Memo 668.

*Rubin, J. (1984) Telling actors from objects. Abstract. WP4, OSA Annual Meeting.

*Rubin, J. & Richards, W. (1984) Color vision: representing material changes. MIT A.I. Lab. Memo 764.

Rubin, J. & Richards, W. (1982) Color vision and image intensities: when are changes material? *Biol. Cybern.*, 45:215 – 226.

*Ullman, S. (1984) Visual Routines". *Cognition*, 18:97 – 159. [Also Center for Cognitive Science Occasional Paper 20 (1984).]

*Yuille, A. & Poggio, T. (1984) Fingerprint theorems. *Proc. Assoc. Artificial Intelligence*, AAAI – 84.

Yuille, A. & Poggio, T. (1983) Fingerprint theorems for zero-crossings. MIT A.I. Lab. Memo 730.

END
8-87
DTIC